



1. Recursive Descent Parsing

Consider a context-free language over the alphabet $\Sigma = \{1, 2, a, b\}$ recognized by the following recursive descent parser:

```
bool term(TOKEN tok) { return *next++ == tok; }
bool S() { TOKEN *save = next; return (next=save, N() && T() && S()) ||
    (next = save, N() && T()); }

//Token for 1 is ONE, 2 is TWO
bool N() { TOKEN *save = next; return ((next = save), term(ONE)) ||
    ((next = save), term(TWO)); }

bool T() { TOKEN *save = next; return (next=save, A()) || (next=save, B()); }

bool A() { TOKEN *save = next; return (next=save, term(ATOK)) ||
    (next=save, T() && term(ATOK)); } // Token for a is ATOK

bool B() { TOKEN *save = next; return (next=save, term(BTOK)) ||
    (next=save, T() && term(BTOK)); } // Token for b is BTOK
```

- Give the context-free grammar from which this parser was produced.
- Write a regular expression that recognizes the same language as the CFG.

2. Give a context-free grammar (CFG) for each of the following languages:

- The set of all strings over alphabet $\{a,b\}$ in the language

$$L : \{a^{2n}b^m a^{3n} | n, m \geq 0\}.$$

- The set of all strings of balanced parentheses, i.e., each left parenthesis has a matching right parenthesis and pairs of matching parentheses are properly nested.
- The set of all nonempty strings over alphabet $\{a,b\}$ in that start and end with the same symbol.

3. Explain why the grammar below is ambiguous.

$$S \rightarrow 0A \mid 1B$$

$$A \rightarrow 0AA \mid 1S \mid 1$$

$$B \rightarrow 1BB \mid 0S \mid 0$$

4. Given the following ambiguous context free grammar

$$S \rightarrow Ab \mid aaB$$

$$A \rightarrow a \mid Aa$$

$$B \rightarrow b$$

- Show that the string $s = aab$ has two leftmost derivations.
- Show the two derivation trees for the string s .
- Find an equivalent unambiguous context-free grammar.
- Give the unique leftmost derivation and derivation tree for the string s generated from your rewritten unambiguous grammar.

Best wishes

Dr. Sherin El Gokhy